Improvements to the Spatial-Domain Lapped Transform in Digital Media Compression

Technical Field

The invention relates generally to block transform-based digital media (e.g., video) compression, and more particularly relates to spatial-domain lapped transforms.

Background

Block Transform-Based Coding

Transform coding is a compression technique used in many audio, image and video compression systems. Uncompressed digital image and video is typically represented or captured as samples of picture elements or colors at locations in an image or video frame arranged in a two dimensional grid. This is referred to as a spatial-domain representation of the image or video. For example, a typical format for images consists of a stream of 24-bit color picture element samples arranged as a grid. Each sample is a number representing color components at a pixel location in the grid within a color space, such as RGB, or YIQ, among others. Various image and video systems may use various different color, spatial and time resolutions of sampling. Similarly, digital audio is typically represented as time-sampled audio signal stream. For example, a typical audio format consists of a stream of 16-bit amplitude samples of an audio signal taken at regular time intervals.

Uncompressed digital audio, image and video signals can consume considerable storage and transmission capacity. Transform coding reduces the size of digital audio, images and video by transforming the spatial-domain representation of the signal into a frequency-domain (or other like transform domain) representation, and then reducing resolution of certain generally less perceptible frequency components of the transform-domain representation. This generally produces much less perceptible degradation of the digital signal compared to reducing color or spatial resolution of images or video in the spatial domain, or of audio in the time domain.

More specifically, a typical block transform-based codec 100 shown in Figure 1 divides the uncompressed digital image's pixels into fixed-size two dimensional blocks (X_1, … X_n), each block possibly overlapping with other blocks. A linear transform 120-121 that does spatial-frequency analysis is applied to each block, which converts the spaced samples within the block to a set of frequency (or transform) coefficients generally representing the strength of the digital signal in corresponding frequency bands over the block interval. For compression, the transform coefficients may be selectively quantized 130 (i.e., reduced in resolution, such as by dropping least significant bits of the coefficient values or otherwise mapping values in a higher resolution number set to a lower resolution), and also entropy or variable-length coded 130 into a compressed data stream. At decoding, the transform coefficients will inversely transform 170-171 to nearly reconstruct the original color/spatial sampled image/video signal (reconstructed blocks $\hat{X}_1, \ldots \hat{X}_n$).

The block transform 120-121 can be defined as a mathematical operation on a vector x of size N. Most often, the operation is a linear multiplication, producing the transform domain output $y = Mx$, M being the transform matrix. When the input data is arbitrarily long, it is segmented into N sized vectors and a block transform is applied to each segment. For the purpose of data compression, reversible block transforms are chosen. In other words, the matrix M is invertible. In multiple dimensions (e.g., for image and video), block transforms are typically implemented as separable operations. The matrix multiplication is applied separably along each dimension of the data.

For compression, the transform coefficients (components of vector y) may be selectively quantized (i.e., reduced in resolution, such as by dropping least significant bits of the coefficient values or otherwise mapping values in a higher resolution number set to a lower resolution), and also entropy or variable-length coded into a compressed data stream.

At decoding in the decoder 150, the inverse of these operations (dequantization/entropy decoding 160 and inverse block transform 170-171) are applied on the decoder 150 side, as show in Fig. 1. While reconstructing the data, the inverse matrix M^{1} (inverse transform 170-171) is applied as a multiplier to the transform domain data. When applied to the transform domain data, the inverse transform nearly reconstructs the original time-domain or spatial-domain digital media.

While compressing a still image (or an intra coded frame in a video sequence), most common standards such as MPEG-2, MPEG-4 and Windows Media partition the image into square tiles and apply a block transform to each image tile. The transform coefficients in a given partition (commonly known as block) are influenced only by the raw data components within the block. Irreversible or lossy operations on the encoder side such as quantization cause artifacts to appear in the decoded image. These artifacts are independent across blocks and produce a visually annoying effect known as the blocking effect. Likewise for audio data, when non-overlapping blocks are independently transform coded, quantization errors will produce discontinuities in the signal at the block boundaries upon reconstruction of the audio signal at the decoder. For audio, a periodic clicking effect is heard.

Several techniques are used to combat the blocking effect – the most popular among these are the deblocking filter that smoothes inter block edge boundaries, and spatial extrapolation that encodes differences between the raw input data and a prediction from neighboring block edges. These techniques are not without their flaws. For instance, the deblocking filter approach is "open loop", i.e. the forward transform process does not take into account the fact that deblocking is going to be performed prior to reconstruction on the decoder side. Besides, both these techniques are computationally expensive.

In order to minimize the blocking effect, cross block correlations can be exploited. One way of achieving cross block correlation is by using a lapped transform as described in H. Malvar, "Signal Processing with Lapped Transforms," Artech House, Norwood MA, 1992. A lapped transform is a transform whose input spans, besides the data elements in the current block, a few adjacent elements in neighboring blocks. Likewise, on the reconstruction side the inverse transform influences all data points in the current block as well as a few data points in neighboring blocks.

For the case of 2-dimensional (2D) data, the lapped 2D transform is a function of the current block, together with select elements of blocks to the left, top, right, bottom and possibly top-left, top-right, bottom-left and bottom-right. The number of data points in neighboring blocks that are used to compute the current transform is referred to as the overlap.

Spatial Domain Lapped Transform

The lapped transform can be implemented in the transform domain, as a step that merges transform domain quantities after a conventional block transform. Else, it can be implemented in the spatial-domain by a pre-processing stage that is applied to pixels within the range of overlap. These two implementations are mathematically related and therefore equivalent.

Figure 2 shows an example of a conventional spatial-domain lapped transform. In the example shown, the overlap is 2 pixels, and two pixels each from the two adjacent blocks shown are pre-processed in pre-processing stage 210. Two pre-processed outputs are sent to each of the blocks for block transform-based coding by codec 100 as in Fig. 1. An inverse of the pre-processing stage is applied at post-processing stage 220 after decoding. With a judicious choice of pre-processing and block transform, a wide range of lapped transforms can be realized.

A key advantage of the spatial domain realization of the lapped transform is that an existing block transform-based codec can be retrofitted with a pre- and post-processing stage to derive the benefits of the lapped transform, i.e., reduced block effect and better compression, using an existing codec framework. Pre-processing 210 and post-processing can be represented as a matrix multiplication as shown in Fig. 3. In the conventional spatial-domain lapped transform 200, the pre-processing and post-processing matrices are inverses of each other, i.e., pre-processing matrix (P_f) and the inverse or post-processing matrix (P_i) multiplied together equal the identity matrix I.

However, there is a critical flaw to the conventional spatial-domain lapped transform that prevents its practical use: the expansion of the range of data subsequent to pre-processing.

More specially, a useful pair of pre- and post-processing matrix operations has the following characteristics:

1. The post processing stage 220 "smoothes" the block boundary – mathematically if post processing is implemented as the matrix multiply $\hat{x} = P_i \hat{y}$, then the matrix P_i has its eigenvalues ≤ 1.
2. Since the pre and post processing operations are inverses in the theoretical design, the pre-processing stage has eigenvalues ≥ 1, i.e., it is *range expansive.*
3. Often, a desirable design rule is to require that a linear ramp across the block edge be converted to a step edge at the block boundary by pre-processing. In the example shown in Figure 4, the pixel values 430-433 of two adjacent blocks in an image initially correspond to a color gradient or linear ramp 410. The pre-processing operation converts these pixel values to lie on a step edge 420. In the post-processing step, the same step edge will be converted back to a linear ramp, thereby also squelching (smoothing) the blocking effect that may result from block transform-based coding.

Point (3) above leads to a very poor spectral behavior for matrices P_f and P_i. This causes tremendous range expansion and prevents practical use of the conventional spatial-domain lapped transform in Figure 2. The problem is further exacerbated in 2 and higher dimensions, since the range expansion is squared or raised to a higher power.

Besides the problem of increased range, the pre- and post-processing steps are typically defined in infinite precision. Since the post-processing stage must be replicated on decoders with different floating point implementations that must match with the "golden" decoder, post-processing is often defined in scaled integer arithmetic. The pre-processing matrix may have entries that are floating point.

Summary

Various improvements to the spatial-domain lapped transform (SDLT) in a digital media codec or compression system address the above-noted flaws of the conventional spatial-domain lapped transform depicted in Figures 1-4.

According to a first improvement, the pre- and post-processing operations (also referred to as "filters") used in the SDLT are not required to be inverses of each other. With a judicious choice of these processes, range expansion is kept to a minimum. In general, the pre-processing operation is "relaxed," whereas the post-processing operation is made more "aggressive".

According to a second improvement, the SDLT can include a range-limiting operation. Subsequent to the relaxed pre-processing operation just described, the data points lying outside a pre-determined expanded range are clipped to within a permissible range.

In a third improvement, the SDLT varies the pre-processing and post-processing operations based on a global quality metric (such as a frame level quantization parameter, QP, in video frames). The SDLT includes a set of one or more pairs of pre- and post-processing filters, and chooses a certain pair of filters from the set based on the global quality metric. In a simple embodiment, the compression system uses a pair of pre- and post-processing filters above a threshold QP, and omits the pre- and post-processing operations below the threshold QP.

Various embodiments of the spatial-domain lapped transform can incorporate separate of these improvements individually, or alternatively two or more of these improvements in combination.

Additional features and advantages of the invention will be made apparent from the following detailed description of embodiments that proceeds with reference to the accompanying drawings.

Brief Description Of The Drawings

Figure 1 is a block diagram of a conventional block transform-based codec in the prior art.

Figure 2 is a block diagram of a spatial-domain lapped transform implemented as pre and post processing operations in combination with the block transform-based codec of Figure 1, also in the prior art.

Figure 3 is a block diagram of a pre-processing operation in the spatial-domain lapped transform of Figure 2, also in the prior art.

Figure 4 is a graph illustrating the result of the pre-processing operation in the spatial-domain lapped transform of Figure 2 on pixels near a block boundary between two example blocks in an image.

Figure 5 is a conceptual view of pre- and post-processing passes over an image for an improved spatial-domain lapped transform in Figure 6.

Figure 6 is a block diagram of a block transform-based codec incorporating the improved spatial-domain lapped transform.

Figure 7 is a block diagram of a quality-based switching of pre-processing and range reduction for the improved spatial-domain lapped transform in the block transform-based codec of Figure 6.

Figure 8 is a block diagram of a suitable computing environment for implementing the block transform-based codec with improved spatial-domain lapped transform of Figure 6.

Detailed Description

The following description is directed to a digital media compression system or codec, which implements an improved spatial-domain lapped transform. For purposes of illustration, an embodiment of a compression system incorporating the improved spatial-domain lapped transform is a digital video compression system, specifically the digital video compression codec of the Microsoft® Windows Media Player (WMP) system. Alternatively, the improvements of the spatial-domain lapped transform also can be incorporated into other digital video compression systems or codecs, such as audio and image codecs, whether implementing the WMP standard media compression or another standard (e.g., MPEG or like other standards).

1. Improved Spatial-Domain Lapped Transform

With reference to Figure 6, an improved spatial-domain lapped transform (SDLT) is incorporated into an existing block transform-based compression system (referred to as codec 600). In one example implementation, the block transform-based compression system 600 is the digital video codec implemented in the Microsoft Windows Media Player (WMP), which is described, *inter alia*, in co-pending U.S. Patent Application No. 10/322,383, entitled "Motion Compensation Loop With Filtering," filed December 17, 2002, the disclosure of which is hereby incorporated by reference. For intra-frame blocks of video, the encoder 610 in the WMP Codec applies a forward transform 620-621 to respective intra-blocks of reference frames of the video, then quantizes and entropy encodes the resulting transform coefficients of the blocks in a quantization/entropy code unit 630. In accordance with the WMP video compression standard, the forward transform 620-621 in the encoder 610 can be an integerized form of the Discrete Cosine Transform (DCT) or other similar transform matrix, as described in co-pending U.S. Patent Application No. 10/376,147, entitled "2-D Transforms For Image And Video Coding," filed February 28, 2003, the disclosure of which is hereby incorporated by reference.

At the decoder 650, the inverse of the operations at the encoder are carried out. A dequantization/entropy decode unit 660 decodes and dequantizes the transform-domain representation of respective blocks, then the inverse block transforms 670-671 are applied to reconstruct the spatial-domain representation of the blocks.

The codec 600 also incorporates the improved SDLT implemented as a pre-processing filter 640 added prior to the forward transform 620-621 and post-processing filter 680 following the inverse transform 670-671. The improved SDLT further includes a range reduction operation 642 (labeled "RR").

The pre-processing filter 640 and post-processing filter 680 in the improved SDLT are not exactly matching transforms (i.e., inverses of each other), as are the pre-processing and post-processing operations in conventional SDLTs described in the Background above. Matching is required if the encoder's input must be reconstructed without loss by the decoder. However, in practical video and image coding schemes, data loss is acceptable and inevitable. Accordingly, for digital media compression context, the improved SDLT drops the requirement of matching pre- and post-processing operations, although it remains desirable to minimize the round-trip loss.

In the improved SDLT, the post-processing filter 680 is made more aggressive. Noise introduced due to quantization can be approximated to be wideband, and this fact can be used to improve the decoder side reconstruction by making the post processing filter more aggressive, since it is a smoothing operation. Doing so further reduces blocking artifacts and lowers distortion. The use of a more aggressive post-processing filter has the downside of increasing the reconstruction error of the compression system. But, within reasonable bounds, this downside of increasing error especially when the true edge is coincident with the block edge is kept manageable.

Likewise, the pre-processing filter 640 in the improved SDLT is made more relaxed from its nominal (i.e., the inverse of the post-processing filter). Mathematically, this means that the eigenvalues of the relaxed pre-processing filter's matrix are smaller than the eigenvalues of the conventional SDLT pre-processing operation. Consequently, range expansion due to pre processing is reduced.

More specifically, the pre-processing filter 640 and post-processing filter 680 in the improved SDLT are defined as follows. The pre-processing and post-processing filters can be implemented as matrix multiplications with matrices, P_f and P_i, respectively, as shown in the following Table 1. In this definition, X is an input overlap block overlaying a border of adjacent transform blocks for the codec, and $\hat{Y}$ is the reconstructed output of the codec for the overlap block. For a conventional SDLT discussed in the Background section above, the matrices P_f and P_i for the pre-processing and post-processing filters are inverses, which is to say the product of the matrices is the identity matrix I. In the improved SDLT, the filters' matrices are mismatched. This means the product of the pre- and post-processing filter's matrices is close-to, but not equal to the identity matrix I. Further, the post-processing filter matrix is made more aggressively smoothing than the nominal post-processing filter in the conventional SDLT (which is an inverse of the pre-processing matrix). The pre-processing filter matrix can then be more relaxed than a nominal inverse of the post-processing matrix, which reduces its range expansion effect. This means the product of the eigenvalues of the pre- and post-processing filter matrices is less than one.

Table 1. Definition of Pre-processing and Post-processing Filters in the Improved SDLT

Pre-processing: $Y = P_f X$

Post-processing: $\hat{X} = P_i \hat{Y}$

Conventional lapped transform / inverse transform pair: $P_i \cdot P_f = I$

Mismatched pair (in current invention): $P_i \cdot P_f \approx I$

$$\text{such that: } \begin{cases} \mathrm{Eig}(P_f) = \sigma_{f0}, \sigma_{f1}, \ldots \geq 1 \\ \mathrm{Eig}(P_i) = \sigma_{i0}, \sigma_{i1}, \ldots \leq 1 \\ \sigma_{fk} \cdot \sigma_{ik} \leq 1 \end{cases}$$

The improved SDLT further includes a range reduction operation 642 following the pre-processing filter 640. Since the matching constraint is removed, both pre and post processing operations can be implemented in fixed point arithmetic. Moreover, range expansion can be limited by clipping the pre-processed output to within a permissible range. Although clipping is nonlinear and lossy, its downside can be reduced by properly choosing the prefilter and clipping thresholds.

The improved SDLT also features varying the pre- and post-processing filter mismatch of the SDLT in relation to signal quality achieved by the codec. More specifically, the magnitude of rate-distortion loss due to mismatched pre and post processing filters varies as a function of the compression quality, which can be expressed as a quality metric, such as the quantization parameter QP in the WMP codec. At higher quantization levels (where QP is larger), the distortion as a result of compression is rather large and the mismatch has little to no detriment. In fact, the aggressive post filter has the salubrious effect of better smoothing over noise. However at very low QPs, the codec

600 operates in the near lossless domain. Here, the mismatch may badly affect the rate-distortion performance.

In the codec 600, the SDLT is varied based on quality by simply disabling the SDLT, and revert to the ordinary block transform when the QP is below a threshold. For quantization levels above the threshold where the rate-distortion loss of mismatched SDLT filters is low compared to quantization loss, the SDLT transform is enabled. In an implementation of the codec adhering to the WMP video standard, the SDLT transform is switched off at QP≤7.

Figure 7 depicts an encoder 700 for an alternative embodiment of the SDLT that further varies the pre- and post-processing filter mismatch relative to compression quality. This alternative embodiment includes a switchable bank of pre and post-processing filters 710-712 with a quality-based switch 730 that chooses a pre-processing filter from the filter bank by a pre-determined or transmitted parameter (i.e., a parameter transmitted between encoder and decoder in the compressed digital media stream) representing a quality metric. A similar switch in the decoder of this embodiment likewise chooses a post-processing filter, such that a pair of pre- and post-processing filters is selected for the given quality level. (The decoder in this embodiment can be a reverse of the illustrated encoder 700, leaving out the range reduction operation 720-722.) In the WMP standard, one such parameter is the quantization parameter (QP) of the video frame. At a very low QP, the pre and post processing steps are disabled (i.e. the corresponding multiplier matrices are identities). At mid levels, the post-processing filter does some moderate amount of smoothing. At high levels of QP, the post-processing filter is a strong smoothing operation. Similarly, the degree of relaxation of the pre-processing filter (and consequent reduction in range expansion of the pre-processing operation) increases with the QP level. In further alternative embodiments, other or separate quality metrics than the quantization parameter can be used to vary the filter mismatch of the SDLT.

2. Illustration of Applying the SDLT to a Video Frame

With reference now to Figure 5, an implementation of the codec 600 for the WMP compression standard divides an image that is to be coded as an intra-frame or reference frame into 8x8 pixel blocks that are the input of the block transform 620-621. The SDLT in the codec 600 achieves an overlapped transform on blocks with an overlap of 2 pixels by first pre-processing along the borders of the blocks with the pre-processing filter. The pre-processing filter 640 in this case operates on a 4x4 block that extends over the border of adjacent 8x8 transform blocks. In the codec 600, the pre-processing filter is applied at the top-left, top, top-right, left, right, bottom-left, bottom and bottom-right of each 8x8 transform block. This can be accomplished by applying the pre-processing filter to successive 4x4 blocks straddling the block boundaries of adjacent 8x8 transform blocks in successive horizontal and vertical passes over the intra-frame image, as illustrated in Figure 5. In this way, the coefficients of each 8x8 transform block after pre-processing is then influenced by a 12x12 block from the original intra-frame image (prior to the pre-processing passes) that is concentric with the 8x8 block. The subsequent block transform of the 8x8 transform block is then actually a transform of coefficients derived from a 12x12 block from the original intra-frame image, which overlaps the adjacent 12x12 blocks of succeeding transforms (except for a 4x4 area in the center of each transform block). In effect, this overlap reduces the blocking artifacts that would result from independently coding 8x8 transform blocks.

In the WMP implementation of the codec 600, the SDLT is used to effect an overlapped transform in encoding both intra blocks in inter-frames, as well as all blocks in intra-frames. The intra-frames are reference frames, which are encoded without reference to other frames in the video's temporal sequence. On the other hand, the inter-frames are encoded with relative to a preceding and/or subsequent intra-frame.

A suitable pre-processing filter preferably is implemented as a matrix of integer values, with a possible scaling by a multiple of two. This allows the pre-processing filter to be more efficiently executed on a processing unit of a computer, or on a graphics or audio processor. An example of a suitable pre-processing filter matrix for the WMP implementation of the codec 600 is the following matrix, P_f.

$$P_f = \begin{bmatrix} 37 & 0 & 0 & -5 \\ 7 & 37 & -5 & -7 \\ -7 & -5 & 37 & 7 \\ -5 & 0 & 0 & 37 \end{bmatrix} /32$$

For input pixel values with a range R, the worst case output range resulting from this pre-processing filter matrix 640 is equal to $1.75R$. For two dimensional data (e.g., image or video), the range expansion factor for this pre-processing filter matrix is $1.75^2 = 3.0625$. In the case of the WMP video standard, the input raw pixel data is expected to be integer values in the range [0,255]. For compression, the pixel input data is re-centered to the range [-128, 127], after which the pre-processing filter 640 is applied.

In the range reduction (clipping) operation 642 (Figure 6), the data points below -256 are clipped to -256, and those above 255 are clipped to 255. In practice, such situations (where the pre-processing filter produces values outside the range) are not common. The range [-256, 255] is represented in 9 bits, which is the expected input values to the forward block transform 620-621 in the WMP standard.

The post-processing filter 680 also is preferably implemented as a matrix of integer values, with a possible scaling by a multiple of 2. An example of such a filter suitable for use in a codec complying to the WMP video standard is the following matrix, P_i.

$$P_i = \begin{bmatrix} 7 & 0 & 0 & 1 \\ -1 & 7 & 1 & 1 \\ 1 & 1 & 7 & -1 \\ 1 & 0 & 0 & 7 \end{bmatrix} /8$$

The post-processing filter 680 is applied to 4x4 block bridging the boundaries of adjacent 8x8 transform blocks similar to the pre-processing filter passes as illustrated in Figure 5. Following the post-processing, the pixels are restored to their original value range by adding an offset of 128.

3. Computing Environment

The above described codec with improved SDLT can be performed on any of a variety of devices in which digital media signal processing is performed, including among other examples, computers; image and video recording, transmission and receiving equipment; portable video players; video conferencing; and etc. The digital media coding techniques can be implemented in hardware circuitry, as well as in digital media processing software executing within a computer or other computing environment, such as shown in Figure 8.

Figure 8 illustrates a generalized example of a suitable computing environment (800) in which described embodiments may be implemented. The computing environment (800) is not intended to suggest any limitation as to scope of use or functionality of the invention, as the present invention may be implemented in diverse general-purpose or special-purpose computing environments.

With reference to Figure 8, the computing environment (800) includes at least one processing unit (810) and memory (820). In Figure 8, this most basic configuration (830) is included within a dashed line. The processing unit (810) executes computer-executable instructions and may be a real or a virtual processor. In a multi-processing system, multiple processing units execute computer-executable instructions to increase processing power. The memory (820) may be volatile memory (e.g., registers, cache, RAM), non-volatile memory (e.g., ROM, EEPROM, flash memory, etc.), or some combination of the two. The memory (820) stores software (880) implementing the described encoder/decoder and transforms.

A computing environment may have additional features. For example, the computing environment (800) includes storage (840), one or more input devices (850), one or more output devices (860), and one or more communication connections (870). An interconnection mechanism (not shown) such as a bus, controller, or network interconnects the components of the computing environment (800). Typically, operating system software (not shown) provides an operating environment for other software executing in the computing environment (800), and coordinates activities of the components of the computing environment (800).

The storage (840) may be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, CD-RWs, DVDs, or any other medium which can be used to store information and which can be accessed within the computing environment (800). The storage (840) stores instructions for the software (880) implementing the codec with improved SDLT.

The input device(s) (850) may be a touch input device such as a keyboard, mouse, pen, or trackball, a voice input device, a scanning device, or another device that provides input to the computing environment (800). For audio, the input device(s) (850) may be a sound card or similar device that accepts audio input in analog or digital form, or a CD-ROM reader that provides audio samples to the computing environment. The output device(s) (860) may be a display, printer, speaker, CD-writer, or another device that provides output from the computing environment (800).

The communication connection(s) (870) enable communication over a communication medium to another computing entity. The communication medium conveys information such as computer-executable instructions, compressed audio or video information, or other data in a modulated data signal. A modulated data signal is a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired or wireless techniques implemented with an electrical, optical, RF, infrared, acoustic, or other carrier.

The digital media processing techniques herein can be described in the general context of computer-readable media. Computer-readable media are any available media that can be accessed within a computing environment. By way of example, and not limitation, with the computing environment (800), computer-readable media include memory (820), storage (840), communication media, and combinations of any of the above.

The digital media processing techniques herein can be described in the general context of computer-executable instructions, such as those included in program modules, being executed in a computing environment on a target real or virtual processor. Generally, program modules include routines, programs, libraries, objects, classes, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The functionality of the program modules may be combined or split between program modules as desired in various embodiments. Computer-executable instructions for program modules may be executed within a local or distributed computing environment.

For the sake of presentation, the detailed description uses terms like "determine," "generate," "adjust," and "apply" to describe computer operations in a computing environment. These terms are high-level abstractions for operations performed by a computer, and should not be confused with acts performed by a human being. The actual computer operations corresponding to these terms vary depending on implementation.

In view of the many possible embodiments to which the principles of our invention may be applied, we claim as our invention all such embodiments as may come within the scope and spirit of the following claims and equivalents thereto.